A Lightweight Security Framework for Wireless Sensor Networks

Tanveer A. Zia
Charles Sturt University
Locked Bag 588, Boorooma St.
Wagga Wagga, NSW 2678, Australia
tzia@csu.edu.au

Albert Y. Zomaya
The University of Sydney
Sydney, NSW 2006
Australia
albert.zomaya@sydney.edu.au

Abstract

Wireless sensor networks are a promising future of many commercial and military applications. However, these networks pose unique security challenges. While the deployment of sensor nodes in an unattended environment makes the networks vulnerable to a variety of potential attacks, the inherent power and memory limitations of sensor nodes makes conventional security solutions unfeasible. Though there has been some development in the field of sensor network security, the solutions presented thus far address only some of security problems faced. The deployment of sensor networks in many sensitive applications requires an ample solution. This paper presents a computationally lightweight security framework to provide a comprehensive security solution against the known attacks in sensor networks. The proposed framework consists of four interacting components: a secure triple-key scheme (STKS), secure routing algorithms (SRAs), a secure localization technique (SLT) and a malicious node detection mechanism. Singly, each of these components can achieve certain level of security. However, when deployed as a framework, a high degree of security is achievable. The framework takes into consideration the communication and computation limitations of sensor networks. While there is always a tradeoff between security and performance, experimental results prove that the proposed framework can achieve high degree of security with negligible overheads.

Keywords: Wireless sensor networks security, security framework, secure key management, secure routing, secure localization, malicious node detection

1 Introduction

Wireless sensor networks have become a promising future to many applications, such as smart houses, smart farms, smart parking, smart hospitals, habitat monitoring, building and structure monitoring, distributed robotics, industrial and manufacturing, and national security. In addition to common network threats, sensor networks are more vulnerable to security breaches because they are physically accessible by adversaries. Imagine the damage caused by compromised sensor network in sensitive military and hospital applications. Many developments have been made in introducing countermeasures to potential threats in sensor networks; however, sensor network security remains less addressed area. Most of the security solutions available in literature address a specific security problem in sensor networks ignoring other issues. This paper collates and extends the security mechanisms [1],[2],[3],[4],[5] to form a comprehensive security framework and provides a computationally lightweight defense against all the known attacks in wireless sensor networks.

1.1 Our Contribution

In form of a security framework we make four contributions in securing sensor networks:

(1) A secure triple-key management scheme (STKS) to mitigate the confidentiality and authentication related attacks.

(2) Secure routing algorithms (SRAs) which address potential threats in node to cluster leader and cluster leader to base station and vice versa communication.

(3) A secure localization technique (SLT) to ensure the safe nodes positions.

(4) A malicious node detection mechanism based on a monitoring mechanism.

1.2 Organization of paper

In Section 2, we summarize related work with some analysis. Section 3 provides a brief discussion on wireless sensor networks, its topology, platform used in our analysis, terms and notations, and possible attacks on sensor networks. We then present the security framework in section 4 that comprises four major components; secure triple key management scheme, secure routing mechanism, secure localization technique, and malicious node detection with the analysis of proposed framework and results comparing with existing security solutions. Finally we conclude our paper in Section 5.

2 Related work

Eschenauer and Gilgor [6] presented a probabilistic key pre-distribution scheme where each sensor node receives a random subset of keys from a large key pool before deployment. To agree on a key for communication, two nodes find one common key within their subsets and use that key as their shared key. Chan et al. [7] extended this idea and developed three key pre-distribution schemes; q-composite, multipath reinforcement, and random-pairwise keys schemes.

Pietro et al. [8] have presented a random key assignment probabilistic model and two protocols; ‘direct and cooperative’ to establish a pairwise communication between sensors by assigning a small set of random keys to each sensor. This idea later converged to pseudo random generation of keys which is energy efficient as compared to previous key management schemes.

Liu and Ning [9] introduced a general framework for establishing pairwise keys between sensors on the basis of a polynomial-based key pre-distribution protocol [10]. Afterwards they presented two instantiations of the general framework: a random subset assignment key pre-distribution scheme, and a hypercube-based key pre-distribution scheme. Finally, they presented a technique to reduce the computation at sensors so that their schemes can be implemented efficiently.

A pairwise key pre-distribution [11] is an effort to improve the resilience of the network by lowering the initial payoff of smaller scale network attacks and pushes adversary to attack at bigger scale to compromise the network. Later in their work Du et al. [12] presented a key scheme based on deployment knowledge. This key management scheme takes advantage of the deployment knowledge where sensor position is known prior to deployment. Because of the randomness of deployment, it is not feasible to know the exact location of neighbors, but knowing the set of likely neighbors is realistic, this issue is addressed using the random key pre-distribution [6].

Zhu et al. [13] have presented LEAP; a security mechanism having a key management scheme based on a set of four keys for each sensor node which restricts the security impact of a node to the immediate neighborhood of the compromised node.

Marti et al. [14] have proposed *watchdog* and *pathrater* tools to detect and mitigate routing behavior, where watchdog detects a misbehaving node, however, the listed weaknesses such as ambiguous collisions, limited transmission power, false misbehavior and collusions make this technique less effective.

Perrig et al. [15] have introduced SPINS (Security Protocols for Sensor Networks). SPINS is a collection of security protocols (SNEP) and mirco-TESLA. SNEP (Secure Network Encryption Protocol provides data confidentiality and two-way data authentication with minimum overhead. Micro-TESLA, a micro version of TESLA (Time Efficient Streamed Loss-tolerant Authentication) provides authenticated streaming broadcast.

SPINS leaves some questions like security of compromised nodes, DoS issues and network traffic analysis issues. Furthermore, this protocol assumes the static network topology ignoring the ad hoc and mobile nature of sensor nodes.

Undercoffer et al. [16] proposed a light weight security protocol that operates in the base station of sensor communication where base station can detect and remove an aberrant node if it is compromised. This protocol does not specify any security measures in case of any passive attacks on node where an adversary is intercepting the communication.

Lazos and Poovendran [17] have proposed a secure localization technique based on the use of directional antennas. Referring to some applications where sensor nodes to be disguised having directional antennas is not feasible. Capkun and Hubaux [18] used explicit RF distance bounding in order to obtain a verifiable location in the presence of attackers. This scheme assumes the known position of certain nodes “landmarks”. Landmarks are placed across the network in an organized manner which we think would be an issue in applications such as battlefield where nodes are deployed by dropping from aircrafts. Therefore, determining landmarks position may not be practical then. Anjum et al. [19] present a secure localisation algorithm based on transmission of nonces at different power levels from anchor nodes. This raises an issue of a node which has exhausted its power.

Beacon Suite [20] is a combination of techniques to detect illegitimate beacon nodes providing incorrect information about the locations of sensor nodes. These techniques include detection of malicious beacon signals, detection of replayed beacon signals, identification of malicious beacon nodes, avoidance of false detection, and revocation of malicious beacon nodes. The beacon nodes are used to provide location information to the sensor nodes as well as to perform detect signals from other beacon nodes.

Beacon Suite uses an *alert* counter to detect the suspicious behavior of a beacon node and a *report* counter to record the number of alerts reported. The robustness of Beacon Suite is questionable because these two counters work on a discrete scale and the revocation process is centralized because discrete scale is not desirable for robustness instead continuous scale and a reputation and trust-based mechanism is more robust solution [21].

A t-degree trivariate symmetric polynomial is predistributed to establish key agreement between pair of nodes in a two-dimensional space [22]. Authors suggest that the use of global polynomial in their scheme can achieve perfect resilience to the node compromise. It will be interesting to see the degree of resilience against multiple attacks in sensor networks. A Just-Enough Redundancy Transmission scheme [23] which uses the Maximum-Distance Separable (MDS) codes is used to reduce the total number of information transmitted and sends the MDS codes to encode the secret link key information through multiple multihop paths. Although this scheme reduces the transmission resources, it still leaves the possibility of adversaries to capture the transmission by capturing all the signals sent through the multiple paths.

3 Wireless Sensor Networks

Wireless sensor networks consist of large number of tiny sensors and actuators with limited energy, computations and transmission power [24],[25]. Sensor nodes are randomly deployed in an environment where they are prone to physical interaction and most likely left unattended after deployment. Although nodes have many limitations but they report to fusion nodes called cluster leaders which transmit data to base station and are believed to be a powerful computer safely located with large computational resources.

3.1 Topology

We consider a hierarchical topology of sensor networks where sensor nodes form a parent child relationship in clusters when deployed [26], [27], [28]. In this topology, nodes broadcast their IDs and listen to the neighbors, add the neighbors IDs in its routing table and count the number of neighbors it could listen to. Hence these connected neighbors become a cluster. Each cluster elects a sensor node as a leader. All inter-cluster communication is routed through cluster leaders. Cluster leaders also serve as fusion nodes to aggregate packets and send them to the base station. A cluster leader receives highest number of messages, this role changes after reaching an energy threshold, hence giving opportunity to all nodes becoming a cluster leader when nodes move around in a dynamic environment. Coverage of cluster depends on the signal strength of the cluster leader. Cluster leader and its neighbor nodes form a parent-child relationship in a tree-based network topology. In this multi hop cluster model, data is collected by the sensor nodes, aggregated by the cluster leader and forwarded to the next level of cluster leader, eventually reaching the base station. Due to the deployment nature, nodes are highly vulnerable to localization attacks from compromised networks and malicious nodes.

3.2 Platform used in our analysis

A typical sensor network contains large number of densely deployed, tiny, low cost nodes that use wireless mesh network. Sensor networks use multi-hop and cluster based routing algorithms based on dynamic network, and resources algorithms based on dynamic network and discovery protocol [29]. For our analysis and simulation we have used MICA2 [30] due to its popularity in wireless sensor networks research community.

3.3 Terms and Notations

Table 1 defines the terms and notations, which are used in this paper.

Table 1: Terms and Notations

ID#	A unique ID of the sensor node
TS	An encrypted time stamp for beacon authentication
Aggr message	Aggregated message by a cluster leader
CL	Cluster leader - a node randomly selected as a leader for a given group of sensors through a leader election process
BS	Base station, a node assumed to be very powerful with extra ordinary computation resources
$MAC_K(m)$	Message authentication code for message m, generated using key k
Level	Level of node - value indicate the number of hops between the base station and node
K_n	Network key (K_n) - generated by the base station, broadcasted in whole sensor network, and shared by the entire sensor network
K_s	Sensor key (K_s) - generated by the base station, based on a seed and sensor ID, pre-deployed in each sensor node and shared by sensor node and base station.
K_c	Cluster key (K_c) - generated by the cluster leader and shared by the nodes in that particular cluster.

3.4 Threat Model

Threats in sensor networks can be classified as external and internal. External threats occur from outside the sensor network and may amount to mere passive eavesdropping on data transmissions, but can extend to injecting bogus data into the network to consume network resources and rage Denial of Service (DoS) attacks. Internal threats stem from compromised nodes running malicious data or from attackers who have stolen the cryptographical contents from legitimate nodes. The proposed framework addresses both internal and external threats.

Roosta et al. [31] have categorised attackers as *mote-class* attackers and *laptop-class* attackers. A mote-class attacker has access to a few motes with the same capabilities as other motes in the network. A laptop-class attacker has access to more powerful devices, such as laptops. This gives the adversary an advantage over the sensor network since it can launch more serious attacks.

An *insider* attack versus an *outsider* attack: An outside attacker has no special access to the sensor network, such as passive eavesdropping, whereas an inside attacker has access to the encryption keys or other codes used by the network. Thus an inside attacker could, for example, be a compromised node which is a legitimate part of the sensor network.

A *passive* attacker versus an *active* attacker: Passive attackers are only interested in collecting sensitive data from the sensor network, which compromises the privacy and confidentiality requirements. In contrast, the active attackers' goal is to disrupt the function of the network and degrade its performance. For example, the attacker might inject faulty data into the network by pretending to be a legitimate node.

3.5 Attacks and Security Concerns

Ideally a network should meet the security goals of CIAA – Confidentiality, Integrity, Authentication and Access control [32]. *Confidentiality* means ensuring a message remains concealed from any attack, *integrity* refers to the trustworthiness of message that it has not been tampered with, *authentication* is confirming that the message is from the node where it claims to be from and *access control* is the ability to determine if a node has access to the right resources. Two major reasons why wireless sensors networks are posing unique security challenges are (1) node constraints: energy, processing power and memory limitations, and (2) network constraints: wireless and ad hoc nature of network. Many attacks [33] have been identified in sensor networks.

Some of the common attacks are summarized below:

- **Selective forwarding:** Selective forwarding is a way to influence the network traffic by believing that all the participating nodes in network are reliable to forward the message. In selective forwarding attack, malicious nodes simply drop certain messages instead of forwarding every message. Once a malicious node cherry picks on the messages, it reduces the latency and deceives the neighboring nodes that they are on a shorter route. Effectiveness of this attack depends on two factors. First the location of the malicious node, the closer it is to the base station the more traffic it will attract. Second is the percentage of messages it drops. When selective forwarder drops more messages and forwards less, it retains its energy level thus remaining powerful to trick the neighboring nodes.

- **Sinkhole attacks:** In sinkhole attacks, adversary attracts the traffic to a compromised node. The simplest way of creating sinkhole is to place a malicious node where it can attract most of the traffic, possibly closer to the base station or malicious node itself deceiving as a base station. One reason for sinkhole attacks is to make selective forwarding possible to attract the traffic towards a compromised node. The nature of sensor networks where all the traffic flows towards one base station makes this type of attacks more susceptible.

- **Sybil attacks:** A type of attacks where a node creates multiple illegitimate identities in sensor networks either by fabricating or stealing the identities of legitimate nodes. Sybil attacks can be used against routing algorithms and topology maintenance; it reduces the effectiveness of fault tolerant schemes such as distributed storage and dispersity. Another malicious factor is geographic routing where a Sybil node can appear at more than one place simultaneously.

- **Wormholes:** In wormhole attacks an adversary positioned closer to the base station can completely disrupt the traffic by tunneling messages over a low latency link. Here an adversary convinces the nodes which are multi hop away that they are closer to the base station. This creates a sinkhole because adversary on the other side of the sinkhole provides a better route to the base station.

- **Hello flood attacks:** Broadcasting a message with stronger transmission power and pretending that the HELLO message is coming from the base station. Message receiving nodes assume that the HELLO message sending node is the closest one and they try to send all their messages through this node. In this type of attacks all nodes will be responding to HELLO floods and wasting the energies. The real base station will also be broadcasting the similar messages but only few nodes will have responding to it.

- **DoS attacks:** Denial of service attacks occur at physical level causing radio jamming, interfering with the network protocol, battery exhaustion etc.

4 The Security Framework

The security framework consists of four components: (1) Secure triple-key management scheme, (2) Secure routing mechanism, (3) secure localization mechanism, and (4) malicious node detection technique. Secure routing and secure localization mechanisms are protected by secure triple key management scheme to ensure communication secrecy and authenticity. If the proposed key management scheme compromises then we can detect the malicious node using our malicious node detection mechanism. Figure 1 illustrates the proposed framework. Following section describes the four components of the framework.

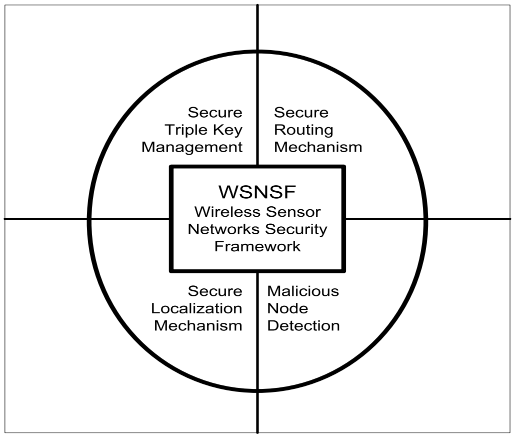

Secure Triple Key Management
Secure Routing Mechanism
WSNSF
Wireless Sensor Networks Security Framework
Secure Localization Mechanism
Malicious Node Detection

Figure 1: Wireless Sensor Networks Security Framework

4.1 The Secure Triple-Key Management Scheme

Key management is critical to meet the security goals of confidentiality, integrity and authentication to prevent the Sensor Networks being compromised by an adversary. Due to ad-hoc nature and resource limitations of sensor networks, providing a right key management is challenging. Traditional key management schemes based on trusted third parties like a certification authority (CA) are impractical due to unknown topology prior to deployment. Trusted CA is required to be present at all times to support public key revocation and renewal [34]. Trusting on a single CA for key management is more vulnerable, a compromised CA will risk the security of entire sensor network. Fei et al. [34] decompose the key management problem into:

- *Key pre-distribution*: installation of keys in each sensor node prior to distribution

- *Neighbour discovery*: discovering the neighbour node

- *End-to-end path key establishment*: end to end communication with those nodes which are not directly connected

- *Isolating aberrant nodes*: identifying and isolating damaged nodes.

- *Key-establishment latency*: reducing the latency resulted from communication and power consumption.

The fundamental problem we realize in wireless sensor network security is to initialize the secure communication between sensor nodes by setting up secret keys between communicating nodes. In general we call this *key establishment*. There are three types of key establishment techniques [11], [12]: trusted-server scheme, self enforcing scheme, and key pre-distribution scheme. The trusted server scheme depends on a trusted server e.g., Kerberos [35]. The self-enforcing scheme depends on asymmetric cryptography using public keys. However, limited computation resources in sensor nodes make this scheme less desirable. A simple solution is to store a master secret key in all the nodes and obtain a new pairwise key. In this case capture of one node will compromise the whole network. Storing the master key in tamper resistant sensor nodes increases the cost and energy consumption of sensors. Another key pre-distribution scheme [11] is to let each sensor carry $N - 1$ secret pairwise keys, each of which is known only to this sensor and one of the other $N-1$ sensors (N is the total number of sensors). Extending the network makes this technique impossible as existing nodes will not have the new nodes keys.

Our secure triple-key management scheme [2] consists of three keys: two pre-deployed keys in all nodes and one in-network generated cluster key for a cluster to address the hierarchical nature of sensor network.

- K_n (network key): Generated by the base station, pre-deployed in each sensor node, and shared by the entire sensor network. Nodes use this key to *encrypt* the data and pass onto next hop.

- K_s (sensor key): Generated by the base station, pre-deployed in each sensor node, and shared by the entire sensor network. Base station uses this key to *decrypt* and process the data and cluster leader uses this key to *decrypt* the data and send to base station.

- K_c (cluster key): Generated by the cluster leader, and shared by the nodes in that particular cluster. Nodes from a cluster use this key to *decrypt* the data and forward to the cluster leader. Nodes will use this key only when they are serving the purpose as a cluster leader, otherwise nodes will not need to decrypt the message received from other nodes thus saving the energy and processing power.

Triple key serves the purpose of confidentiality and authentication. Section below describes how this scheme works:

4.1.1 Base station to node key calculation

Base station uses K_n to encrypt and broadcast data. When a sensor node receives the message, it decrypts it by using its K_s. This process follows as: Base station encrypts its own ID, a current time stamp *TS* and its K_n as a private key. Base station generates a random seed *S* and assumes itself at level 0. The packet contains following fields:

K_n	MAC	ID	TS	message	Level 0

Sensor node decrypts the message received from the base station using K_s. Here MAC is message authentication code for a message (m).

4.1.2 Nodes to Cluster leader key calculation

When node sends a message to cluster leader, it constructs the message as follows:

$$\{\mathrm{ID}_{sn}, K_n, \mathrm{TS}, \mathrm{MAC}, (\mathrm{message})\}$$

Cluster leader checks the ID from the packet, if the ID in the packet matches the ID it holds, it verifies the authentication and integrity of the packet through MAC. Otherwise, packet is dropped by the cluster leader. Node builds the message using the fields below:

K_n	MAC	ID	TS	message	Level 2

Figure 2 illustrates the key calculation process from nodes to cluster leaders and to the base station.

4.1.3 Cluster leader to next hop cluster leader key calculation

Cluster leader aggregates the messages received from its nodes and forwards it to next level cluster leader or if the cluster leader is one hop away from the base station, it directly sends the message to the base station. Receiving cluster leader checks its routing table and constructs the following packet to be sent to next level cluster leader or to the base station. Cluster leader adds its own ID *CLn*, its network and cluster key in incoming packet and rebuilds the packet as under:

$$\mathrm{ID}, K_{CLn}, [\mathrm{ID}_{sn}, K_n, \mathrm{TS}, \mathrm{MAC}, (\mathrm{Aggr\ message})]\}$$

K_n	MAC	ID	TS	message	Level 1

Here ID is the ID of receiving cluster leader which wraps the message and sends it to the next hop cluster leader or to the base station if directly connected. Next hop cluster leader receives the packet and checks the ID, if the ID embedded in the packet is same as it holds, it updates the ID for the next hop and broadcast it, or else the packet is discarded. *Aggr message* refers to the message aggregated by the cluster leader.

4.1.4 Cluster leader to base station key calculation

Base station receives the packet from its directly connected cluster leader; it checks the ID of sending cluster leader, verifies the authentication and integrity of the packet through MAC. Cluster leader directly connected with base station adds its own ID along with the packet received from the sending cluster leader. Packet contains the following information:

$$\mathrm{ID}_{CL2}[\mathrm{ID}_{CL1}, K_n, [\mathrm{ID}_{s2}, K_n, \mathrm{TS}, \mathrm{MAC}, (\mathrm{Aggr\ message})]]\}$$

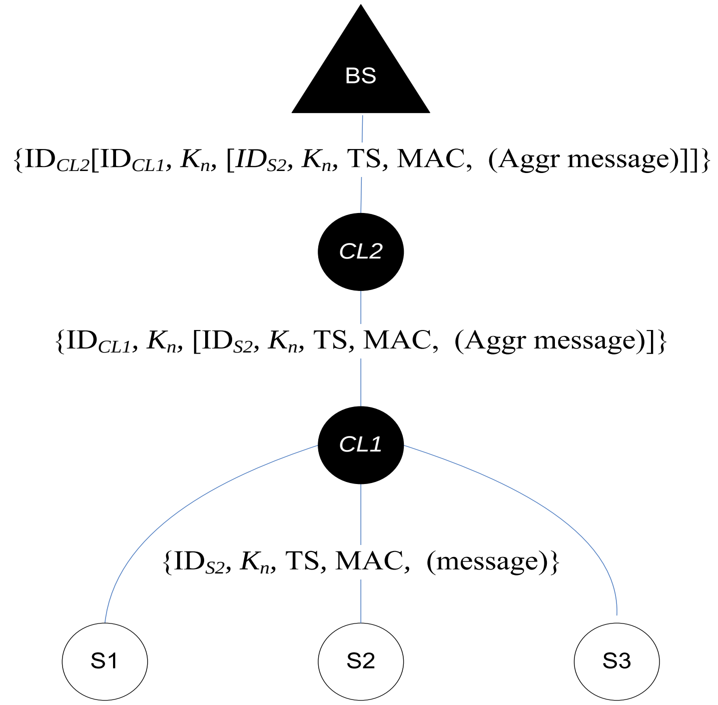

BS
$\{ID_{CL2}[ID_{CL1}, K_n, [ID_{S2}, K_n, TS, MAC, (Aggr message)]]\}$
CL2
$\{ID_{CL1}, K_n, [ID_{S2}, K_n, TS, MAC, (Aggr message)]\}$
CL1
$\{ID_{S2}, K_n, TS, MAC, (message)\}$
S1
S2
S3

Figure 2: Key calculation from Sensor Node S2 to Cluster Leader *CL*1, Cluster Leader *CL*1 to Cluster Leader *CL*2, and Cluster Leader *CL*2 to the Base Station BS

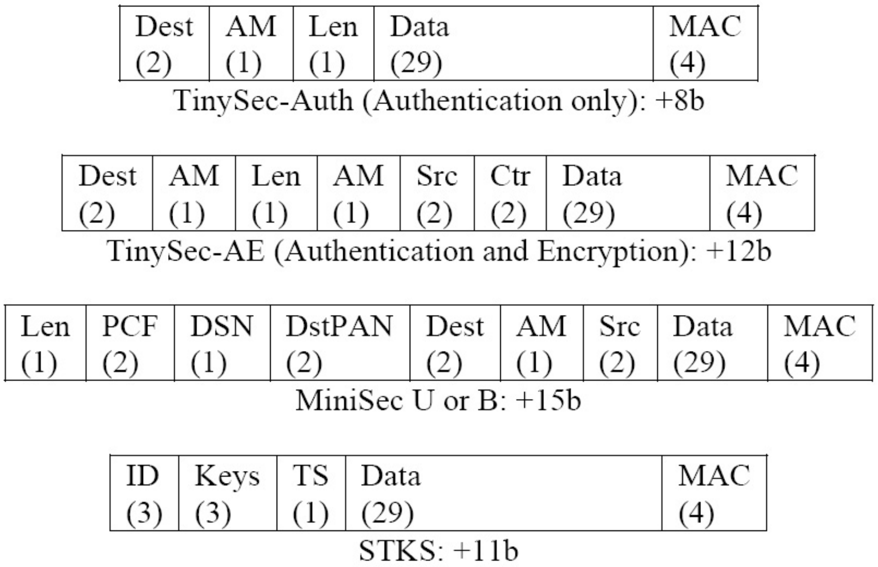
Table 2: Comparison of overheads in TinySec and WSNF

	Application Data (b)	Packet Overhead (b)	Total Size (b)	Time to transmit (ms)	Increase over TinyOS stack	Latency Overhead	Energy Overhead
TinySec-Auth	29	8	37	26.6	1.50%	1.70%	3%
TinySec-AE	29	12	41	28.8	8%	7.30%	10%
STKS	29	11	40	28.3	6.30%	5.90%	8.20%

4.1.5 Analysis of Secure Triple Key Management Scheme

To analyze performance and overheads of the proposed Secure Triple Key Scheme (STKS) it was compared with two well known security schemes TinySec[36] and MiniSec [37]. The packet format comparison in Figure 3 and overheads comparison in Table 2 shows that the STKS does not have any additional overheads. Also it overcomes the weaknesses of TinySec. As per our analysis TinySec is confusing because of its three different states: (1) no TinySec (CRC), (2) TinySec-Auth and (3) TinySec-AE. Also TinySec assumes a message length of 8 bytes or more and does not address smaller messages. TinySec fails to provide secure localization or a secure routing mechanism while STKS address these weaknesses. Furthermore, TinySec does not provide security against insider attacks when a node is captured or compromised, whereas STKS detects the malicious nodes and disperses the information about the presence of such malicious nodes to their neighbours. Although MiniSec provides a good level of its overheads are far greater than those for TinySec and for STKS.

Figures 4 (a)-(b) shows a comparative study of the security overheads in TinySec, MiniSec and STKS. It can be observed that the overheads for STKS are lower than those of the other three security schemes as can be seen from the experimental results and evaluation of other components of the proposed



Dest	AM	Len	Data	MAC
(2)	(1)	(1)	(29)	(4)

TinySec-Auth (Authentication only): +8b

Dest	AM	Len	AM	Src	Ctr	Data	MAC
(2)	(1)	(1)	(1)	(2)	(2)	(29)	(4)

TinySec-AE (Authentication and Encryption): +12b

Len	PCF	DSN	DstPAN	Dest	AM	Src	Data	MAC
(1)	(2)	(1)	(2)	(2)	(1)	(2)	(29)	(4)

MiniSec U or B: +15b

ID	Keys	TS	Data	MAC
(3)	(3)	(1)	(29)	(4)

STKS: +11b

Figure 3: Packet format in TinySec-Auth, TinySec-AE, MiniSec and STKS

framework. Low overheads allow the effective use of RC5 and CBC to achieve high degree of security in sensor networks.

4.2 Secure Routing

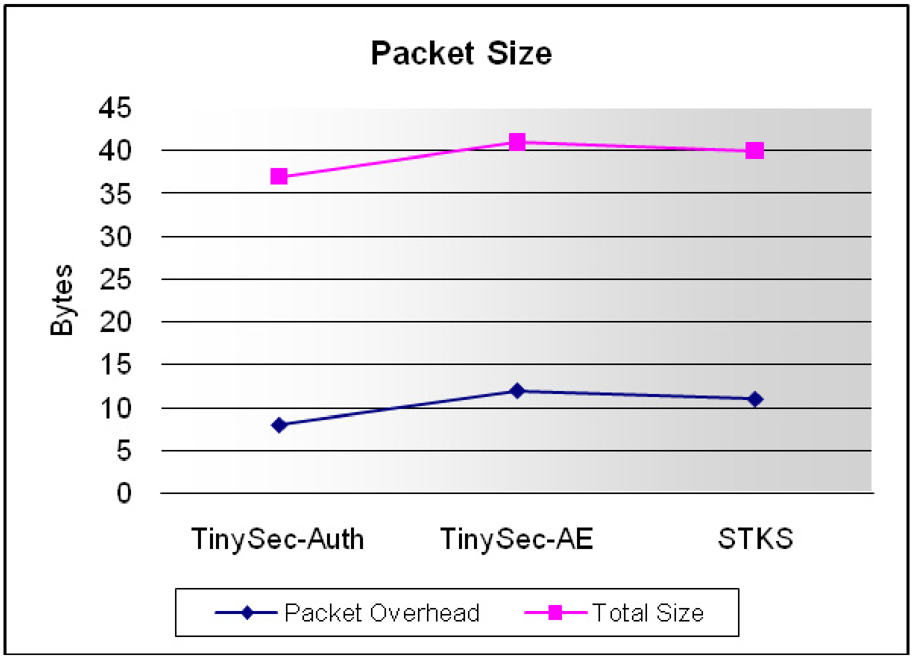
In our secure routing mechanism[1] all the nodes have a unique ID#. Once the network is deployed, base station builds a table containing ID#s of all the nodes in the network. After self organizing process base station knows the topology of the network. Nodes use our secure triple-key management scheme to collect the data, pass onto the cluster leader which aggregates the data and sends it to the base station. We adapt the energy efficient secure data transmission algorithms by[38] and modify it with our secure triple-key management scheme to make it resilient against attacks in wireless sensor networks. Two algorithms (1) sensor node and (2) base station are presented below for secure data transfer from node to base station and base station to node communication:

Node algorithm performs the following functions:

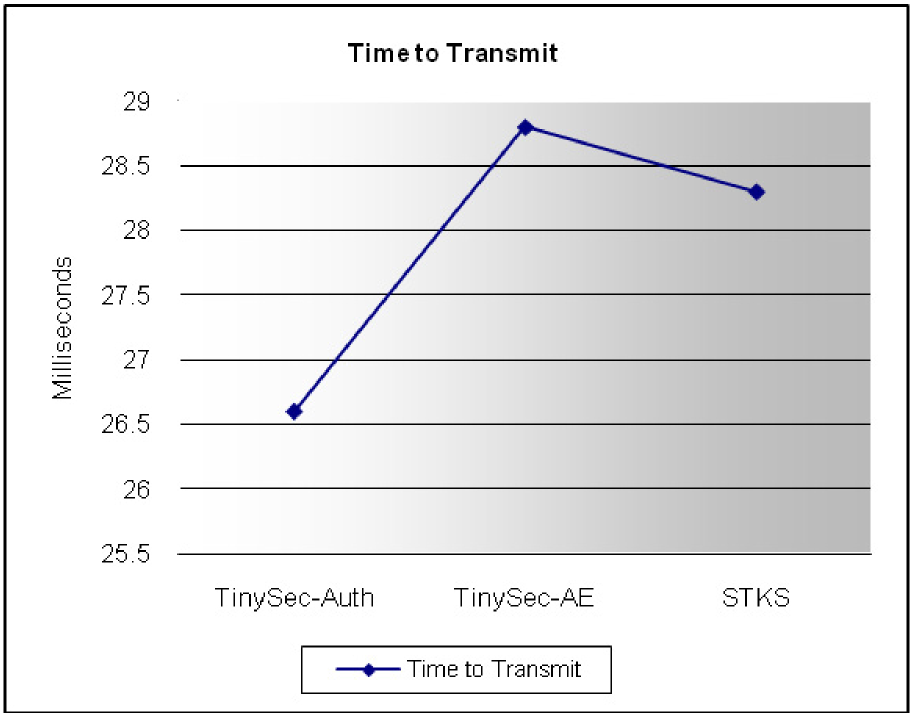
- Sensor nodes use the K_n to encrypt and transmit the data
- Transmission of encrypted data from nodes to cluster leader
- Appending ID# to data and then forwarding it to higher level of cluster leaders
- Cluster leader uses K_c to decrypt and then uses its K_n to encrypt and send the data to next level of cluster leaders, eventually reaching the base station

Base station algorithm is responsible of following tasks:

- Broadcasting of K_s and K_n by the base station
- Decryption and authentication of data by the base station



(a) Packet comparison



(b) Time to transmit a packet

Figure 4: TinySec vs STKS

4.2.1 Node algorithm

- Step 1: If sensor node *i* wants to send data to its cluster leader, go to step 2, else exit the algorithm

- Step 2: Sensor node *i* requests the cluster leader to send K_c.

- Step 3: Sensor node *i* uses K_c and its own K_n to compute the encryption key $K_{i,cn}$.

- Step 4: Sensor node *i* encrypts the data with $K_{i,cn}$ and appends its ID# and the TS to the encrypted data and then sends them to the cluster leader.

- Step 5: Cluster leader receives the data, appends its own ID#, and then sends them to the higher-level cluster leader or to the base station if directly connected. Go to Step 1.

Figure 5 demonstrates this algorithm and illustrates the communication between sensor node *i* and the cluster leader.

4.2.2 Base Station Algorithm

- Step 1: Check if there is any need to broadcast the message. If so, broadcast the message encrypting it with K_n.

- Step 2: If there is no need to broadcast the message then check if there is any incoming message from the cluster leaders. If there is no data being sent to the base station go to step 1.

- Step 3: If there is any data coming to the base station then decrypt the data using K_s, ID# of the node and TS within the data.

- Step 4: Check if the decryption key K_s has decrypted the data perfectly. This leads to check the credibility of the TS and the ID#. If the decrypted data is not perfect discard the data and go to step 6.

- Step 5: Process the decrypted data and obtain the message sent by sensor nodes

- Step 6: Decide whether to request all sensor nodes for retransmission of data. If not necessary then go back to step 1.

- Step 7: If a request is necessary, send the request to the sensor nodes to retransmit the data. When this session is finished go back to step 1.

This routing technique using our triple-key management scheme provides a strong resilience towards spoofed routing information attacks, selective forwarding, sinkhole attacks; Sybil attacks wormholes and HELLO flood attacks presented in [33].

4.3 Secure Localization

Determining the location of nodes is very important for many sensitive applications. Due to the deployment nature of sensor networks security is a major concern. This section is divided into two parts (1) determining the node location (2) securing node location.

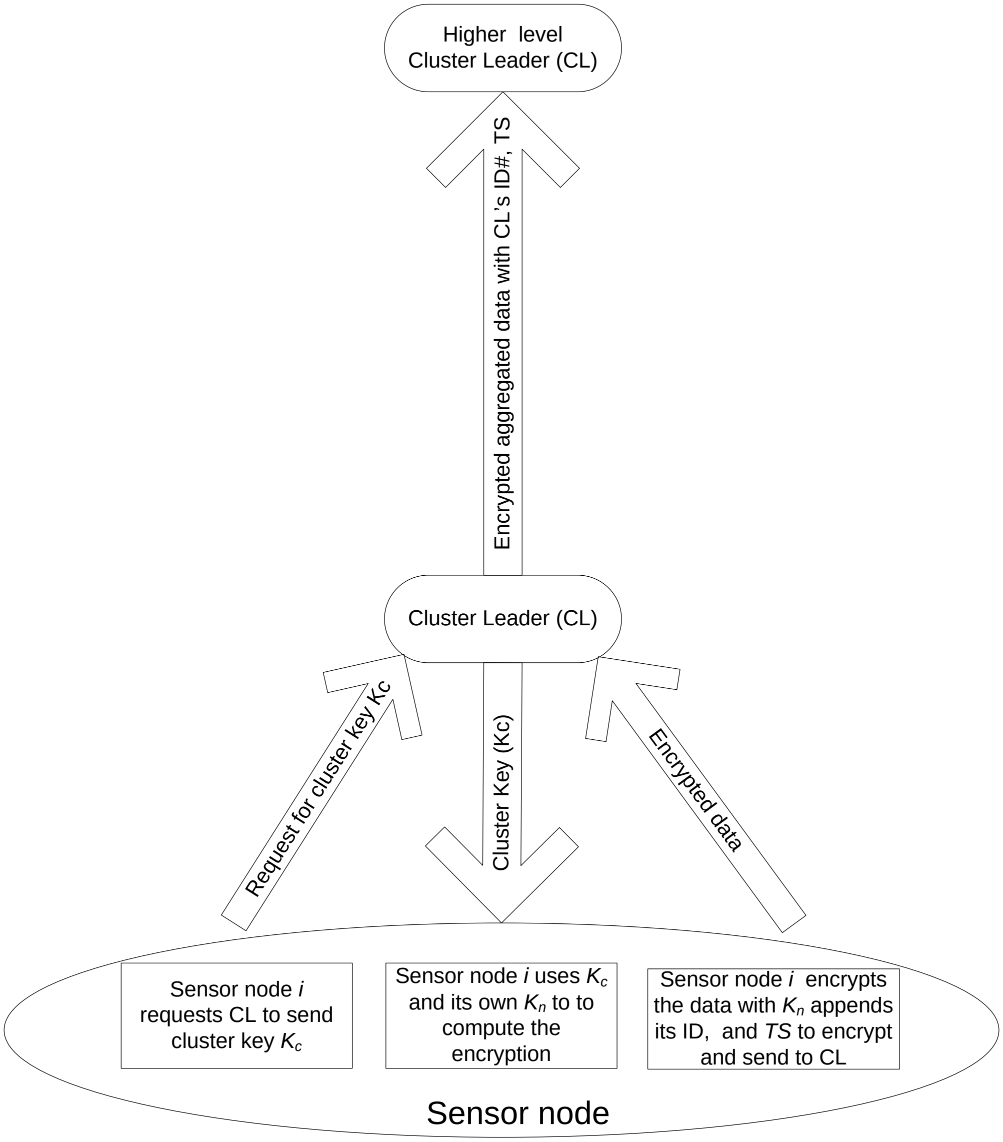
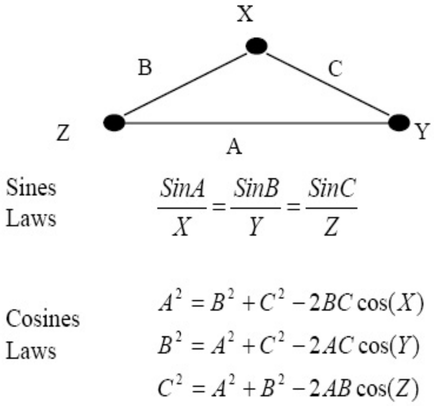

Higher level Cluster Leader (CL)
Encrypted aggregated data with CL's ID#, TS
Cluster Leader (CL)
Request for cluster key Kc
Cluster Key (Kc)
Encrypted data
Sensor node *i* requests CL to send cluster key K_c
Sensor node *i* uses K_c and its own K_n to to compute the encryption
Sensor node *i* encrypts the data with K_n appends its ID, and *TS* to encrypt and send to CL
Sensor node

Figure 5: Sensor node *i* to cluster leader and base station communication

4.3.1 Determining the Node Location

A basic feature of a location system is the ability to determine the location of a node and verify its distance from the neighboring nodes [39]. In our secure Localization mechanism each node determines its position by calculating its distance from its neighbours using well known four methods in Triangulation: Lateration, attenuation, propagation and angulations. Figure 6 (a)-(d) illustrate the triangulation process to determine the node location. Each node determines its position by calculating its distance from its neighbours. Node location in triangulation is calculated by using trigonometry laws of sines and cosines as follows:

- **Lateration:** We assume that when the nodes are deployed they know their location through an atomic multileteration [40] process. In this process node estimates its location if it is in the range of three other nodes. When a base station sends beacon to form the network topology, nodes reply with their position in the network. Each node determines its position by calculating its distance


X
B
C
Z
A
Y
Sines Laws
SinA/X = SinB/Y = SinC/Z
Cosines Laws
A² = B² + C² − 2BC cos(X)
B² = A² + C² − 2AC cos(Y)
C² = A² + B² − 2AB cos(Z)

from its neighbours.

- **Attenuation:** In attenuation triangulation model, signal strength decreases as distance between two node increases. We assume a dense network where nodes are deployed in close distances. In a hierarchical clustered model parent nodes are aware of their child nodes locations.
- **Propagation:** Node A sends a message to node B, node B calculates the time difference $t2 - t1$ between two nodes, $t1$ is the time recorded when a message leaves Node A and $t2$ is when a message arrives at Node B.
- **Angulations:** Angulations use angles to determine the distance between nodes using directional antennas. In 2D position two angles and one distance measurement is used, while in 3D position two angles, one length and one azimuth measurement is used.

4.4 Securing the Node Location – An analysis

Nodes change their position when they move in a dynamic network or if an adversary has compromised the node. In the event of compromise a node is considered a malicious node. The localization process described here is protected by the secure triple-key management scheme.

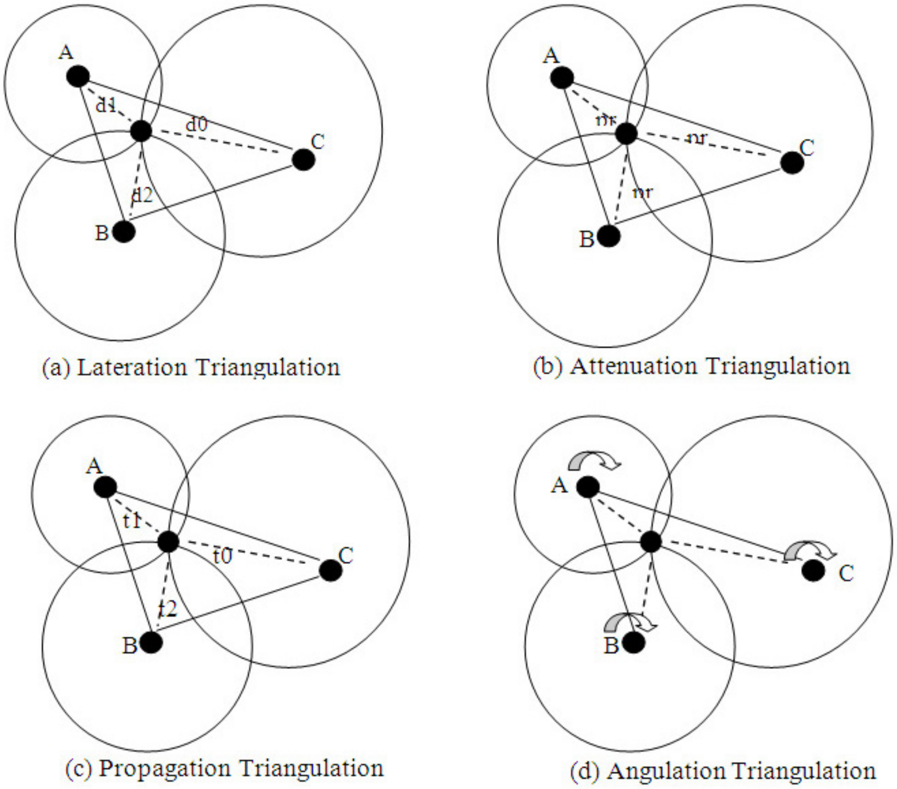
The base station broadcasts a beacon message to the sensor network; this message is encrypted by K_n. If the receiving node is a cluster leader it decrypts the message using K_s and encrypts it again with its K_n and forwards it to the nodes in its cluster. Every node in that cluster use its K_c to decrypt the message, adds its location and reply back to the cluster leader with its location encrypted with K_n. Cluster leader receives the locations from all nodes in the cluster and encrypts it with K_n and sends it to the base station. Base station uses its K_s to decrypt the message and becomes aware of the nodes location in the entire network. The process of base station to cluster leader and nodes and vice versa is described in the following steps:

- Step 1: To establish the secure communication, base station builds a packet which contains:

  ID$_{BS}$, K_n, TS, MAC, S (message)
- Step 2: Cluster leader builds a packet containing the following information:

  ID$_{CL}$, K_n, TS, MAC, S (message)
- Step 3: Nodes to cluster leader packet consists of:

  ID$_{sn}$, K_n, TS, MAC, S (message)



(a) Lateration Triangulation

(b) Attenuation Triangulation

(c) Propagation Triangulation

(d) Angulation Triangulation

Figure 6: Triangulations

- Step 4: Cluster leader aggregates the messages received from the nodes in its cluster and forwards it to the base station using the packet: ID$_{CL}$, K_n, TS, MAC, S (Aggr message).

4.5 Malicious Node Detection Mechanism

This section describes the fourth and last component of the framework. In our malicious node detection mechanism we consider the dynamic and scalable nature of sensor networks where sensor nodes are replaced after reaching energy exhaustion. Message sending node observes the packet receiving node hence becoming a monitor to watch the behavior of receiving node. Due to broadcast nature of wireless sensor networks the monitoring node watches if the receiving node is sending the packet intact or alters the packet contents other than adding its header information.

A malicious node is a compromised node where an adversary has somehow able to break the encryption [24] and has got access to the secure keys and routing protocols of the sensor network. Malicious node detection mechanism is protected by our underlying security framework which is based on a set of three secure keys (TKS) This section demonstrates how a malicious node is detected if in a less likely event of secure triple key management scheme compromise.

In the proposed malicious node detection technique we use a monitoring mechanism. In this mechanism when a node A sends message to node B, it converts itself to a monitoring mode we refer here as A_m. Due to the broadcast nature of wireless sensor networks A_m monitors the behavior of node B after sending the message. When node B transmits the message to the next node, A_m hears that and compares with the message it has sent to node B, hence establishing *original* and *actual* message. If the message transmitted by node B is *original* then node A_m ignores it and continues with its own tasks but if there

Table 3: Node Suspicious Table

Node ID	Suspicious entries	Unsuspicious entries
ID	$NS > 1$	$NU > 1$

is a difference between *original* and *actual* messages greater than a predefined threshold, the message is considered as suspicious and node B is now considered as a suspicious node B_s. In our experimental evaluation we used a value of 3 as threshold to determine an anomaly.

Each node builds a *node suspicious* table containing the reputation of nodes in the cluster. Entries in this table contain the node ID, and the number of suspicious and unsuspicious entries. Nodes update this table every time they identify a suspicious activity by increasing suspicious count by one for that particular node. In Table 3 below, ID is the unique ID of sensor node; *NS* denote node suspicious and *NU* is the node unsuspicious entries.

All the nodes locally build a *node suspicious* table. Every time A_m identifies a suspicious entry it adds into its node suspicious table and disseminate this information among neighbors and all the nodes listening to this message update their *node suspicious* table. This broadcast message also act as an inquiry. Nodes listening to this message reply with their opinion about B_s. In Figure 7, Nodes C and D are neighboring nodes of A_m and B_s, they listen the transmission from B_s and respond with suspicious entry if the suspicious count for B_s in its node suspicious table is greater than its unsuspicious count, otherwise it responds with unsuspicious. Figure 8 (a) shows a message sent by Node A, secured with our network key K_n and in Figure 8 (b), an altered message is shown from Node B.

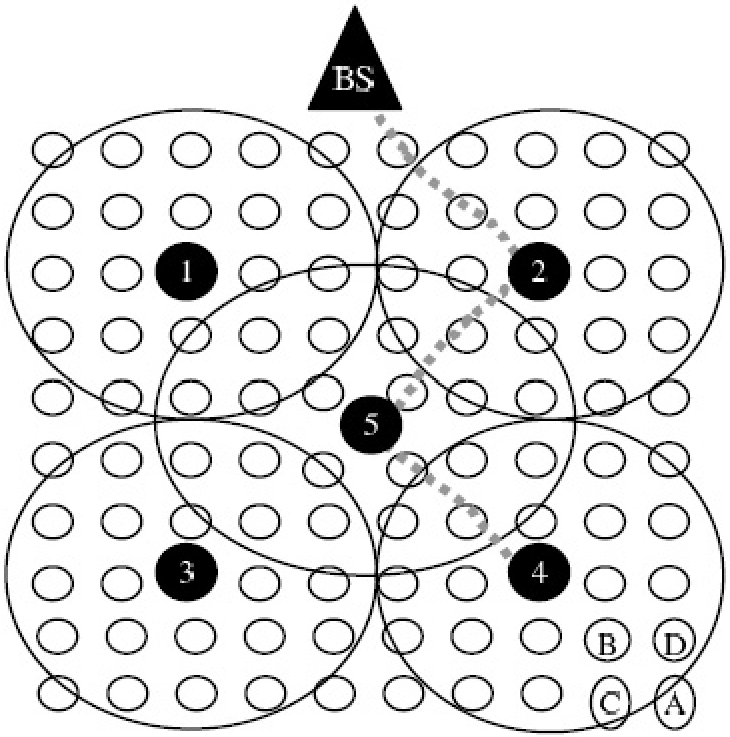
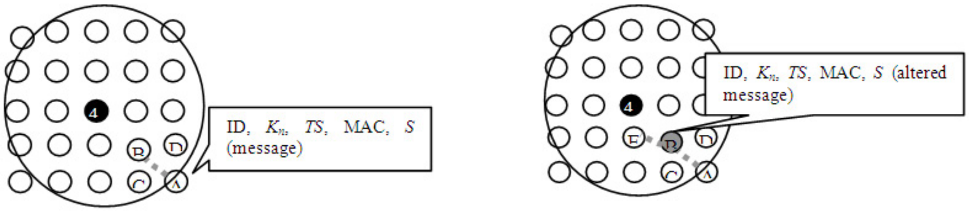


Figure 7: Node *Am* (monitoring node) *Bs* (Suspicious node) and Nodes *C* & *D* neighboring nodes.

ID is the node's unique identifier, K_n is the network key, *TS* is an encrypted time stamp, MAC is the message authentication code generated using K_n for message m and S is the randomly generated seed


4
A
B
C
D
ID, K_n, TS, MAC, S (message)
4
A
B
C
D
E
ID, K_n, TS, MAC, S (altered message)

(a) Message sent by Node A
ID, K_n, TS, MAC, S (message)

(b) Message altered by Node B
ID, K_n, TS, MAC, S (altered message)

Figure 8: Messages sent by nodes A and B

value by the base station.

Node A_m collects the replies from neighbors and updates its *node suspicious* table; it increases its own suspicious entry for B_s by one and the unsuspicious entries accordingly.

Once the *suspicious* entries reach a threshold, node A_m broadcasts that node B_s is a *suspicious* node and all the neighboring nodes update their *node suspicious* tables that a malicious node is present in the cluster. When the presence of a *suspicious* node message reaches a Cluster Leader, it isolates B_s by erasing B_s ID from its *nodes table* and discards any message coming from B_s. Cluster leader broadcasts the message that node B_s has been isolated, therefore any message originated from B_s is discarded by its neighboring nodes hence isolating node B_s from the network.

4.5.1 Experimental Evaluation

This section presents an evaluation of how well the proposed malicious node detection scheme performs in a multi-hop network. J-SIM was used to simulate malicious node detection. Starting with a scenario of 100 nodes randomly deployed over an area of 100 x 100 metres, a node transmission range of 30m was assumed. One of the nodes was to randomly become malicious. The scheme works as follows:

Neighbouring nodes assess a malicious node by monitoring the actual and sent values of data. Whenever any node detects a malicious neighbour, it increases its suspicious node counter by 1 and broadcasts a message to inform other neighbouring nodes. Whenever the counter reaches a threshold of 3 for a specific node, its neighbours consider that node malicious.

The sending node stays awake until the receiving node has forwarded the packet. Because of interference, this scenario might not work all the time; therefore, nodes receive a trust value from their neighbours, the threshold for which can be increased or decreased depending on the application.

Each node transfers one packet every 100 seconds. When a node receives a packet not intended for it, it first checks the destination to see whether it is for one of the neighbouring nodes. If not, it discards the packet. The probability that the node stays awake to monitor its neighbours is 50%. If a malicious node is detected, the detecting node broadcasts the ID of the malicious node to its neighbours.

Once the base station has received the alert about a malicious node from at least 3 neighbouring nodes, it declares the node malicious and isolates it from the network. The base station waits for the alerts from 3 nodes to ensure that the malicious node itself is not generating an alert about the legitimate nodes.

The level of this scheme's security depends entirely on the application. The percentage of neighbours being awake all the time could be 100 percent thus providing complete security. Instead, in order to be more energy efficient, the topology works by letting each node go to sleep when it is not sending or receiving a packet.

As seen from the experimental results shown in Figure 9, the time required to detect a malicious node decreases when the number of nodes in the network is increased. This is because in dense network, the probability of node detection is higher and faster because there are more neighbours monitoring the nodes. The results in Figure 9 are an average of 10 runs.

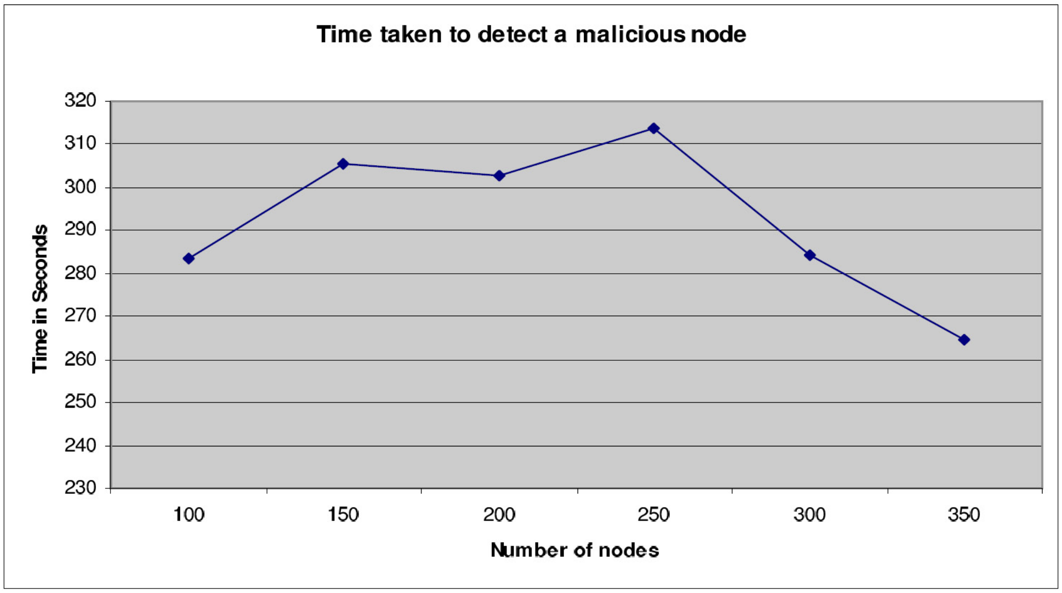


Figure 9: Time taken to detect a malicious node

5 Conclusion

In this paper we presented a computationally lightweight wireless sensor network security framework which is composed of four components: (1) secure triple-key management scheme, (2) secure routing mechanism, (3) secure localization technique, and (4) malicious node detection mechanism. The secure routing mechanism presented ensures a secure node to base station and vice versa communication. We have presented a triple-key management scheme based on two network pre-deployed keys and one cluster deployed key. Triple keys mitigate the confidentiality and authentication related attacks. Localization mechanism presented addresses location determination issues from security perspectives. Lastly the malicious node detection mechanism protects the network from insiders and outsider adversaries. The presented analysis shows that the proposed framework as a whole addresses the security issues competently without increasing the overheads. In contrast to the computationally extensive security solutions, the framework has great potential for emerging applications. Results presented show the effectiveness of the framework to ensure the total security for wireless sensor networks by reducing the packet transmission time, low latency and less packet overheads.

References

[1] T. A. Zia and A. Y. Zomaya, “A security framework for wireless sensor networks,” in *Proc. of IEEE Sensor Applications Symposium (SAS’06), Houston, Texas, USA*. IEEE, February 2006, pp. 49–53.

[2] ——, “A secure triple-key management scheme for wireless sensor networks,” in *Proc. of the 25th IEEE International Conference on Computer Communications (INFOCOM’06), Barcelona, Spain*. IEEE, April 2006, pp. 1–2.

[3] ——, "Secure localization in wireless sensor networks," in *Proc. of the 4th IASTED Asian Conference on Communication Systems and Networks (AsiaCSN'07), Phuket, Thailand*. ACTA Press, April 2007, pp. 177–180.

[4] ——, "A malicious node detection mechanism in wireless sensor networks," in *Proc. of the International Conference on Network Security (ICONS'07), Erode, India*. Macmillan Publishers India Ltd., January 2007.

[5] ——, "Quality of security through triple key scheme in wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 10, no. 5, pp. 722–732, May 2010.

[6] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proc. of the 9th ACM conference on Computer and communications security (CCS'02), Washington, DC, USA*. ACM, November 2002, pp. 41–47.

[7] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proc. of the 2003 IEEE Symposium on Security and Privacy (SP'03), Oakland, California, USA*. IEEE, May 2003, pp. 197–213.

[8] R. Di Pietro, L. V. Mancini, and A. Mei, "Random key-assignment for secure wireless sensor networks," in *Proc. of the 1st ACM workshop on Security of ad hoc and sensor networks (SASN'03), Fairfax, Virginia, USA*. ACM, October 2003, pp. 62–71.

[9] D. Liu, P. Ning, and R. Li, "Establishing pairwise keys in distributed sensor networks," *ACM Transactions on Information and System Security*, vol. 8, no. 1, pp. 41–77, February 2005.

[10] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-secure key distribution for dynamic conferences," in *Proc. of the 12th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO'92), Santa Barbara, California, USA, LNCS*, vol. 740. Springer-Verlag, August 1992, pp. 471–486.

[11] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," *ACM Transactions on Information and System Security*, vol. 8, no. 2, pp. 228–258, May 2005.

[12] W. Du, J. Deng, Y. Han, S. Chen, and P. K. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," in *Proc. of The 23rd Conference of the IEEE Communications Society (INFOCOM'04), Hong Kong, China*. IEEE, March 2004, pp. 586–597.

[13] S. Zhu, S. Setia, and S. Jajodia, "LEAP: efficient security mechanisms for large-scale distributed sensor networks," in *Proc. of the 10th ACM Conference on Computer and Communications Security (CCS'03), Washington DC, USA*. ACM, October 2003, pp. 62–72.

[14] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proc. of the 6th annual international conference on Mobile computing and networking (MobiCom'00), Boston, Massachusetts, USA*. ACM, August 2000, pp. 255–265.

[15] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "Spins: security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521–534, September 2002.

[16] J. Undercoffer, S. Avancha, A. Joshi, and J. Pinkston, "Security for sensor networks," in *Proc. of 2002 CADIP Research Symposium (CADIP'02), Baltimore, USA*, October 2002.

[17] L. Lazos and R. Poovendran, "SeRLoc: secure range-independent localization for wireless sensor networks," in *Proc. of the 3rd ACM workshop on Wireless security (WiSe'04), Philadelphia, Pennsylvania, USA*. ACM, 2004, pp. 21–30.

[18] S. Čapkun and J.-P. Hubaux, "Secure positioning of wireless devices with application to sensor networks," in *Proc. of the 24th Conference of the IEEE Communications Society (INFOCOM 2005), Miami, USA*, vol. 3. IEEE, March 2005, pp. 1917–1928.

[19] F. Anjum, S. Pandey, and P. Agrawal, "Secure localization in sensor networks using transmission range variation," in *Proc. of the 2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'05), Washington, DC, USA*. IEEE, November 2005, pp. 194–203.

[20] D. Liu, P. Ning, and W. Du, "Detecting malicious beacon nodes for secure location discovery in wireless sensor networks," in *Proc. of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05), Columbus, Ohio, USA*. IEEE, June 2005, pp. 609–619.

[21] A. Srinivasan, J. Teitelbaum, and J. Wu, “DRBTS: Distributed reputation-based beacon trust system,” in *Proc. of the 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing (DASC’06), Purdue University, Indianapolis, USA*. IEEE, September 2006, pp. 277–283.

[22] Y. Zhou and Y. Fang, “A two-layer key establishment scheme for wireless sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 6, no. 9, pp. 1009–1020, September 2007.

[23] J. Deng and Y. S. Han, “Multipath key establishment for wireless sensor networks using just-enough redundancy transmission,” *IEEE Transactions on Dependable and Secure Computing*, vol. 5, no. 3, pp. 177–190, July-September 2008.

[24] I. Khalil, S. Bagchi, and C. Nina-Rotaru, “DICAS: Detection, diagnosis and isolation of control attacks in sensor networks,” in *Proc. of the 1st International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM’05), Athens, Greece*. IEEE, September 2005, pp. 89–100.

[25] C.-Y. Chong and S. P. Kumar, “Sensor networks: Evolution, opportunities, and challenges,” *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, August 2003.

[26] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *Proc. of the 33rd Annual Hawaii International Conference on System Sciences (HICSS’00), Maui, Hawaii, USA*, vol. 2. IEEE, January 2000, pp. 8020–.

[27] M. Tubaishat, J. Yin, B. Panja, and S. Madria, “A secure hierarchical model for sensor network,” *ACM SIGMOD Record*, vol. 33, no. 1, pp. 7–13, March 2004.

[28] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, “Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks,” *Wireless Networks*, vol. 8, no. 5, pp. 481–494, September 2002.

[29] R. Anderson, H. Chan, and A. Perrig, “Key infection: smart trust for smart dust,” in *Proc. of the 12th IEEE International Conference on Network Protocols (ICNP’04), Berlin, Germany*. IEEE, October 2004, pp. 206–215.

[30] MEMSIC Inc., “MICA2: Wireless Measurement System,” Mica2 Datasheets, http://www.memsic.com/support/documentation/wireless-sensor-networks/category/7-datasheets.html.

[31] T. Roosta, S. Shieh, and S. Sastry, “Taxonomy of security attacks in sensor networks and countermeasures,” in *Proc. of the 1st IEEE International Conference on System Integration and Reliability Improvements (SIRI’06), Hanoi, Vietnam*, December 2006, pp. 13–15.

[32] C. P. Pfleeger and S. L. Pfleeger, *Security in Computing (4th Edition)*. Prentice Hall PTR, 2006. [Online]. Available: http://portal.acm.org/citation.cfm?id=1177321

[33] C. Karlof and D. Wagner, “Secure routing in wireless sensor networks: Attacks and countermeasures,” *Ad Hoc Network Journal*, vol. 1, no. 2-3, pp. 293–315, September 2003.

[34] F. Hu, J. Ziobro, J. Tillett, and N. K. Sharma, “Secure wireless sensor networks: Problems and solutions,” *Journal of Systemics, Cybernetics and Informatics*, vol. 1, no. 4, pp. 90–100, 2003.

[35] B. C. Neuman and T. Ts’o, “Kerberos - an authentication service for computer networks,” *IEEE Communications Magazine*, vol. 32, no. 9, pp. 33–38, 1994.

[36] C. Karlof, N. Sastry, and D. Wagner, “Tinysec: a link layer security architecture for wireless sensor networks,” in *Proc. of the 2nd international conference on Embedded networked sensor systems (SenSys’04), Baltimore, Maryland, USA*. ACM, November 2004, pp. 162–175.

[37] M. Luk, G. Mezzour, A. Perrig, and V. Gligor, “Minisec: a secure sensor network communication architecture,” in *Proc. of the 6th international conference on Information processing in sensor networks (IPSN’07), Cambridge, Massachusetts, USA*. ACM, April 2007, pp. 479–488.

[38] H. Cam, H. Cam, D. Muthuavinashiappan, S. Ozdemir, and P. Nair, “Energy efficient security protocol for wireless sensor networks,” in *Proc. of the 58th IEEE Semiannual Vehicular Technology Conference (VTC2003-Fall), Orlando, Florida, USA*, vol. 5. IEEE, October 2003, pp. 2981–2984.

[39] N. Sastry, U. Shankar, and D. Wagner, “Secure verification of location claims,” in *Proc. of the 2nd ACM workshop on Wireless security (WiSe’03), San Diego, California, USA*. ACM, September 2003, pp. 1–10.

[40] A. Savvides, C.-C. Han, and M. B. Strivastava, “Dynamic fine-grained localization in ad-hoc networks of sensors,” in *Proc. of the 7th annual International Conference on Mobile Computing and Networking (MobiCom’01), Rome, Italy*. ACM, July 2001, pp. 166–179.

Tanveer A. Zia is a *senior lecturer in Computing* at the School of Computing & Mathematics, Faculty of Business, Charles Sturt University. He has earned his PhD in early 2008 from the School of Information Technologies, University of Sydney, supervised by Professor Albert Zomaya. Tanveer's broader research interests are in *information system and network security*. Specifically he is interested in security for handheld devices such as wireless sensor networks, RFID, mobile phones and PDAs. He is also interested in biometric security, cyber security, information assurance, protection against identity theft, trust management, forensic computing, law and ethics in information system security. Tanveer received his Bachelors of Science in Computer Sciences BSCS from Southwestern University, Philippines in 1992, MBA from Preston University USA in 1997, and Master of Interactive Multimedia MIM from University of Technology Sydney in 2004. Tanveer also holds various industry certifications from Microsoft and Cisco. He has published in several international conferences, symposiums and workshops. Tanveer is a *Senior Member* Australian Computer Society and *Certified Professional (MACS Snr CP)*, *Senior Member Institute of Electrical and Electronics Engineers (IEEE), Senior Member International Association of Computer Sciences and Information Technology (IACSIT), Member IEEE Computer Society*, and *Member Australian Information Security Association (AISA)*.

Albert Y. Zomaya is currently the *Chair Professor of High Performance Computing & Networking* and Australian Research Council Professorial Fellow in the School of Information Technologies, The University of Sydney. He is also the Director of the *Centre for Distributed and High Performance Computing* which was established in late 2009. Professor Zomaya held the *CISCO Systems Chair Professor of Internetworking* during the period 2002–2007 and also was Head of school for 2006–2007 in the same school. Prior to that he was a Full Professor in the Electrical and Electronic Engineering Department at the University of Western Australia, where he also led the Parallel Computing Research Laboratory during the period 1990-2002. He served as Associate-, Deputy-, and Acting-Head in the same department, and held visiting positions at Waterloo University and the University of Missouri-Rolla. He is the author/co-author of 6 books, more than 370 publications in technical journals and conferences, and the editor of 9 books and 11 conference volumes. He is currently an associate editor for another 19 journals including some of the leading journals in the field, such as, *IEEE Transactions on Parallel and Distributed Systems* and *Journal of Parallel and Distributed Computing.* He is the Founding Editor of the *Wiley Book Series on Parallel and Distributed Computing* and the Co–Editor (with Professor Yi Pan) of the *Wiley Book Series on Bioinformatics* and (with Professor Mary Eshaghian-Wilner) the *Wiley Book Series on Nature Inspired Computing*. He is the Editor–in–Chief of the *Parallel and Distributed Computing Handbook (McGraw-Hill, 1996)*. Professor Zomaya was the Chair the *IEEE Technical Committee on Parallel Processing* (1999-2003) and currently serves on its executive committee. He has been actively involved in the organization of national and international conferences. He received the *1997 Edgeworth David Medal* from the Royal Society of New South Wales for outstanding contributions to Australian Science. In September 2000 he was awarded the *IEEE Computer Society's Meritorious Service Award.* Professor Zomaya is a chartered engineer (CEng), a Fellow of the IEEE, a Fellow of the Institution of Electrical Engineers (U.K.), and a Distinguished Engineer of the ACM. His research interests are in the areas of high performance computing, parallel algorithms, networking, and bioinformatics.